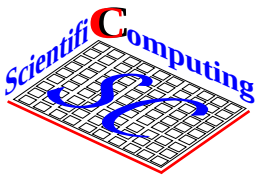
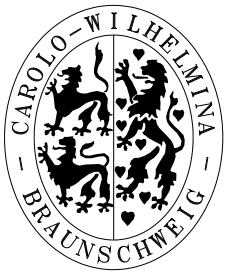


Partitioned but Strongly Coupled Iteration Schemes for Nonlinear Fluid-Structure Interaction

Hermann G. Matthies, Jan Steindorf
Institute of Scientific Computing
Technical University Braunschweig
Brunswick, Germany

Informatikbericht Nr.: 2002-05

October 2002



Partitioned but Strongly Coupled Iteration Schemes for Nonlinear Fluid-Structure Interaction

Hermann G. Matthies, Jan Steindorf
Department of Computer Science
Technical University Braunschweig
Brunswick, Germany

Informatikbericht Nr.: 2002-05

October 2002

Contributed to a special Issue of *Computers and Structures*.

Revised and expanded version of a contribution to the Conference on Computational Mechanics, Leuwen, Belgium, September 2000

Location
Institute of Scientific Computing
Technische Universität Braunschweig
Hans-Sommer-Strasse 65
D-38106 Braunschweig

Postal Address
Institut für Wissenschaftliches Rechnen
Technische Universtität Braunschweig
D-38092 Braunschweig
Germany

Contact
Phone: +49-(0)531-391-3000
Fax: +49-(0)531-391-3003
E–Mail: wire@tu-bs.de

Partitioned but Strongly Coupled Iteration Schemes for Nonlinear Fluid-Structure Interaction

Hermann G. Matthies Jan Steindorf
Institute of Scientific Computing
Technische Universität Braunschweig
Brunswick, Germany
October 2002

Abstract

We look at the computational procedure of computing the response of a coupled fluid-structure interaction problem. We use the so called strong fluid-structure coupling — a totally implicit formulation. At each time step in an implicit formulation, new values for the solution variables have to be computed by solving a nonlinear system of equations, where we assume that we have solvers for the subproblems. This is often the case, when we have existing software to solve each subproblem separately, and want to couple both. We show how to solve the overall nonlinear system by using only the solvers for the subproblems. This is achieved not by considering the equilibrium equations, but the fixed-point problem resulting from the solution iteration for each of the subproblems.

1 Introduction

The numerical simulation of fluid-structure interaction problems is one of the great challenges in scientific computing. Typical examples for fluid-structure interaction arise in aero-elasticity [8], where air flow around an elastic aircraft or oscillations of air-foils in air flow are computed, or in bio-mechanics [4] where the elastic behaviour of micro-pumps or artificial membranes in blood flow is considered. Often, fluid-structure interaction problems are highly nonlinear coupled problems.

The computation of those nonlinear fluid-structure interaction problems requires the simultaneous solution of the strongly coupled fluid and structural equations of motion. The coupling of the fluid and the structure occurs at the fluid-structure interface where kinematic and dynamic coupling conditions are imposed. One of the problems in the computation of fluid-structure interaction is the usually differing formulation of the fluid (Eulerian) and the structure equations (Lagrangian) and therefore the treatment of the moving boundary in the fluid domain.

In this paper, we consider partitioned methods [8] for fluid-structure interaction, i.e. separate solvers are used for the fluid and the structure. Partitioned procedures allow the use of well established discretisation and solution methods for each subproblem. They also support the reuse of existing and highly developed software and offer a modular solution approach.

In partitioned methods [9], the coupled problem is computed with a solution procedure where the fluid and the structure are separately solved and exchange data in every time-step or iteration of the coupling algorithm. There exist various procedures how to couple the fluid and structure solvers: the coupling conditions and the moving interface can be treated in a fully explicit or implicit or in a mixed explicit/implicit manner. This approach allows a smooth transition between "loose" and "strong" coupling. For stability reasons, often a fully implicit formulation has to be used [7]. In this approach, we have to solve a large system of nonlinear equations with the use of the (iterative) solvers for the subsystems. Usually, this is done with Block-Jacobi, Block-Gauss-Seidel or related relaxation methods [3]. Nevertheless, there is a demand for more sophisticated solution methods as the simple methods do not always converge. We will introduce here an approximative Block-Newton method which is shown to be superior to the standard Block-iterative methods [1].

2 Formulation of the Problem

Suppose that a fluid-structure interaction problem has to be solved, where the fluid — described in Eulerian coordinates — occupies the domain Ω_f, and the structure or solid body — described in Lagrangian coordinates — occupies the domain Ω_s. They have the common boundary $\partial\Omega_i = \partial\Omega_f \cap \partial\Omega_s$ where the interaction takes place. On the remaining part of the boundaries $\partial\Omega_f \setminus \partial\Omega_i$ and $\partial\Omega_s \setminus \partial\Omega_i$ we shall assume that appropriate boundary conditions have been specified, which make the whole problem well-posed.

In the spatial fluid domain Ω_f we then have the equations of conservation of momentum

$$\frac{\partial}{\partial t}(\varrho v) + \operatorname{div}(\varrho v \otimes v) = \operatorname{div}\sigma_f + \varrho\beta_f, \tag{1}$$

where ϱ is the density and v the velocity of the fluid, and β_f a force field in the fluid (e.g. gravity); and in addition some constitutive assumption for the fluid stress σ_f, e.g. like for a Newtonian fluid,

$$\sigma_f = -pI + \frac{2}{3}\eta(\operatorname{div} v) + 2\eta D, \tag{2}$$

where p is the pressure, η the viscosity, and $D = \frac{1}{2}(\nabla \upsilon + (\nabla \upsilon)^T)$ the symmetric part of the velocity gradient, together with the incompressibility condition $\operatorname{div} \upsilon = 0$. This then together leads to the incompressible Navier-Stokes equation.

In the material or reference structural domain Ω_s we have the equilibrium equations

$$\frac{\partial}{\partial t}(\rho\varphi) = \operatorname{div}\Sigma_s + \rho\beta_s, \tag{3}$$

where ρ is the density of the structure in the reference configuration, φ the displacement, Σ_s the Piola-Kirchhoff stress, and β_s a force field in the structure (e.g. gravity), together with some constitutive assumption, e.g. $\Sigma_s = \Sigma_s(\nabla\varphi)$ like elasticity.

On the interface we have the conditions of equality of displacement, i.e. the interface $\partial\Omega_i$ is at the position $\xi + \varphi(\xi, t)$ at time t, where ξ is the material reference position. In addition, the velocity has to be equal for corresponding points on the interface, i.e. $\frac{\partial\varphi(\xi,t)}{\partial t} = \upsilon(\xi + \varphi(\xi, t), t)$. In addition to these kinematic conditions, we have to have force equilibrium across the interface $\sigma_f(\xi + \varphi(\xi, t), t) \cdot \nu = -\Sigma_s(\xi) \cdot \nu$, where ν is the normal vector on $\partial\Omega_i$.

3 The Discrete Problem

We assume that the domains for the structure and the fluid have been discretised appropriately, and look at the discrete equations in each sub-domain. We also assume that each of the subproblems has to be solved in a time-implicit manner, and for stability reasons, we would also like a time-implicit procedure for the overall time step. Let us denote the discretised vector of velocities in the fluid by v, the corresponding pressures again by p, the discretised displacement vector in the structure by u and the discretised vector of structure velocities by $\dot{u}$. Then, going from time step n to time step $n + 1$ we have to solve the discretised incremental Navier-Stokes equations

$$N(v^{(n+1)}, p^{(n+1)}, u^{(n+1)}, \dot{u}^{(n+1)}) = 0 \tag{4}$$

in the fluid domain, the discrete form of Eq. (1). For the structure the discrete form of Eq. (3), the discretised incremental structure equilibrium equations

$$S(u^{(n+1)}, \dot{u}^{(n+1)}) = h(v^{(n+1)}, p^{(n+1)}) \tag{5}$$

have to be solved in the structural domain. Only the variables to be solved for at the time step $n+1$ have been displayed for the sake of brevity, everything else (exterior influences and loading, values of the solution variables at the previous time

step n etc.) is assumed to be known and thus hidden away in the equations. We also see the explicit dependence of the fluid equation Eq. (4) on the displacements $u^{(n+1)}$ and velocities $\dot{u}^{(n+1)}$ of the structure at the common boundary at time level $n+1$ (fully implicit formulation, no staggering), and the loading caused by the fluid on the structure as $h(v^{(n+1)}, p^{(n+1)})$ also depending on the fluid velocities and pressures at time level $n+1$ in Eq. (5).

We assume also that for each of the subproblems we have a solver, as is often the case when we have existing software for each of the subproblems separately, and instead of writing new software from scratch for the combined problem, we want to combine this existing software into a new software solving the coupled problem by using the existing solvers as building blocks. For generality we will additionally assume that it is iterative. If one of the solvers is in fact a direct solver, we will view this as a very efficient iterative solver — it only needs one iteration. Let us write the iteration as a fixed point equation:

$$(v_{k+1}^{(n+1)}, p_{k+1}^{(n+1)}) = F(v_k^{(n+1)}, p_k^{(n+1)}, u^{(n+1)}, \dot{u}^{(n+1)}), \qquad k = 0, 1, 2, \ldots$$

for the fluid, where the subscript k is the iteration counter, and

$$(u_{k+1}^{(n+1)}, \dot{u}_{k+1}^{(n+1)}) = G(v^{(n+1)}, p^{(n+1)}, u_k^{(n+1)}, \dot{u}_k^{(n+1)}), \qquad k = 0, 1, 2, \ldots;$$

for the structure; where in this basic form the variables from the "other" domain do not take part in the iteration and have therefore been designated with a tilde as being constant during the iteration. As we are only considering this single time step, we will from now on drop the time step counter n, and we will collectively call the fluid variables $x = (v, p)$, and the structure variables $y = (u, \dot{u})$. Then the above iteration scheme may be written as

$$x_{k+1} = F(x_k, y) \tag{8}$$

for the fluid, and as

$$y_{k+1} = G(x, y_k) \tag{9}$$

for the structure. We do not assume anything specific about these iteration operators, except that each single iteration is a convergent process.

4 Re-Formulation of the Coupled Problem

What we are seeking is a solution of the combined system

$$f(x, y) = x - F(x, y) = 0, \tag{10}$$

$$g(x, y) = y - G(x, y) = 0, \tag{11}$$

which is just a rewritten form of the fixed point form Eq. (8), Eq. (9). Note that these are not derived from the original equilibrium equations Eq. (4), Eq. (5) per time step, but from the iterative solvers for each subproblem Eq. (6), Eq. (7).

An often used form of solving these equations is by Block-Jacobi or Block-Gauss-Seidel iterations [3]:

1. $x_{k+1} = F^{\kappa_1}(x_k, y_k)$, κ_1 iterations of the fluid solver,
2. $y_{k+1} = G^{\kappa_2}(x_{k(+1)}, y_k)$, κ_2 iterations of the structure solver,
3. Check for convergence.

As is well known, the convergence of these methods is often too slow, or they may not be convergent at all, as we have only assumed each single iteration to be convergent, and faster methods — like the Newton-Raphson methods or variants thereof — require the evaluation of derivatives. We want to apply a Block-Newton method, where we approximate the derivatives by finite differences and the use of the iterative solvers F and G along the lines of [1].

One step of the basic Block-Newton method for the combined system Eq. (10)), Eq. (11) entails the solution of the following linear system at each step:

$$\begin{pmatrix} D_x f(x_k, y_k) & D_y f(x_k, y_k) \\ D_x g(x_k, y_k) & D_y g(x_k, y_k) \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y_k \end{pmatrix} = - \begin{pmatrix} f(x_k, y_k) \\ g(x_k, y_k) \end{pmatrix}$$

where $\Delta x_k := x_{k+1} - x_k$, and similarly $\Delta y_k := y_{k+1} - y_k$.

We only want to use the existing solvers, i.e. the iteration mappings F and G. In particular, we do not have direct access to the partial derivatives in Eq. (12). But if we solve the system Eq. (12) by an iterative method, all we need is a way to compute the product of the Jacobian matrix in Eq. (12) by an arbitrary vector.

5 Iterative Solution of the Coupled Problem

As we are now only talking about a single iteration, we will drop the iteration indices, and only look at computing the vector $(\Delta x, \Delta y)^T$ from the right hand side $-(f, g)^T$. To start with, we use – symbolically – Block-Gauss elimination on the system Eq. (12):

$$\begin{aligned} & [D_x f]\Delta x = -[D_y f]\Delta y - f \\ \Longrightarrow \quad & \Delta x = -[D_x f]^{-1} f - [D_x f]^{-1}[D_y f]\Delta y, \end{aligned}$$

and by inserting this into the second equation, we obtain

$$\begin{aligned} S\Delta y &:= ([D_y g] - [D_x g][D_x f]^{-1}[D_y f])\Delta y \\ &= -g + [D_x g][D_x f]^{-1} f =: -r. \end{aligned}$$

In this way we can solve the second equation for Δy, and with this solve the first equation for Δx. We rewrite this as

$$\begin{aligned} S\Delta y &:= ([D_y g] - [D_x g]C)\Delta y \\ &= -g - [D_x g]q =: -r, \end{aligned} \tag{13}$$

with $q := -[D_x f]^{-1} f$ and $C := [D_x f]^{-1}[D_y f]$.

One step of this Newton-Raphson iteration may now be formulated as

1. Solve $[D_x f]q = -f$ for q.
2. Calculate the modified right-hand-side
$$r = g + [D_x g]q.$$
3. Solve $S\Delta y = -r$ for Δy with the Schur complement
$$S = [D_y g] - [D_x g]C.$$
4. Compute $\Delta x = q - C\Delta y$.

It remains to specify how the Jacobians are computed, and how linear systems with the Jacobian matrix $D_x f$ are solved.

In the first step we remember, that we have an iterative solver for the equation with f, and that this step can be seen as one iteration of the Newton-Raphson method for the solution p of the equation $f(x_k + q, y_k) = 0$ when x_k and y_k are fixed. This is just the solution of the fluid equation if the structure variables are known. So we use the iterative solver F here, to obtain $q \approx z_m - x_k$, where

$$z_{j+1} = F(z_j, y),\ j = 0, 1, \ldots, m-1,\ m \geq 1$$

with $z_0 = x_k$. In the second step we use finite differences:

$$r = g + [D_x g]q \approx g(x_k + q, y_k).$$

In the third step we use an iterative method to solve the system with the Schur complement matrix S; say Bi-CGStab or GMRES and so we only need the action of S on some other vector w, again approximated via finite differences with some (small) step-size h:

$$\begin{aligned} \frac{1}{h}S(hw) &= \frac{1}{h}([D_y g](hw) - [D_x g]C(hw)) \\ &\approx \frac{1}{h}(g(x_k - C(hw), y_k + hw) - g(x_k, y_k)). \end{aligned}$$

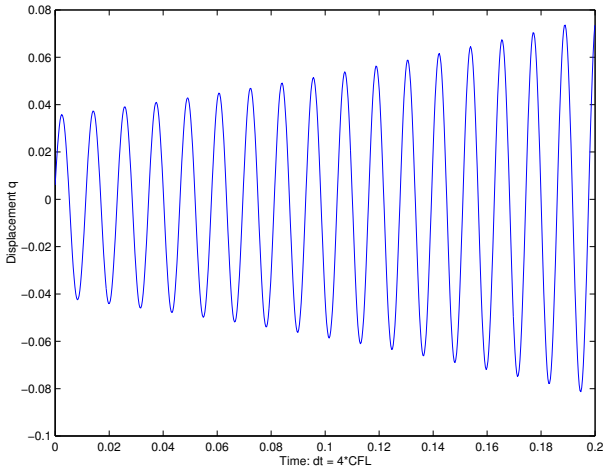


Figure 1: *Displacements of the piston (staggering algorithm)*

For this we also need $C(hw) =: s$, the solution of

$$[D_x f]s = [D_y f](hw),$$

computed as in the first step, where additionally finite differencing will be used for $[D_y f](hw)$:

$$[D_y f](hw) \approx f(x_k, y_k + hw) - f(x_k, y_k).$$

In the fourth step we know q, and again we need the action of C on some vector, this time Δy, known from the third step. This may be done as in the third step for $C(hw)$.

6 Examples

To demonstrate the procedure, we take a simple one-dimensional example [5]: Burger's equation, representing the fluid:

$$\frac{\partial v}{\partial t} + \alpha v \frac{\partial v}{\partial x} - \eta \frac{\partial^2 v}{\partial x^2} = 0 \quad \text{in } (0, \gamma(t)) \tag{14}$$

with $\gamma(0) = 1$, coupled to a piston with mass m on an elastic spring with stiffness k, representing the structure:

$$m\ddot{u} + ku = g \qquad \text{in } (1, 2). \tag{15}$$

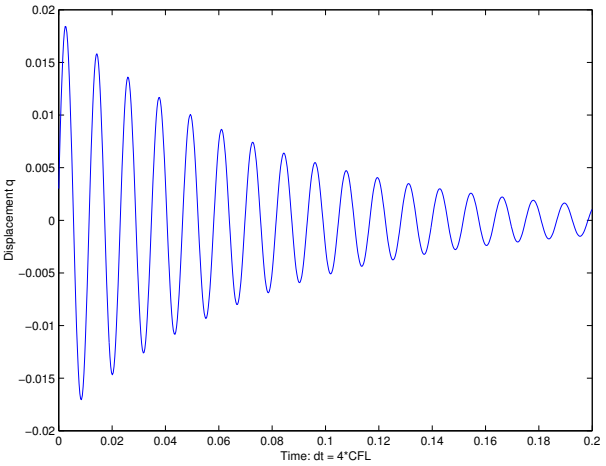


Figure 2: *Displacements of the piston (monolithical algorithm)*

The coupling conditions at the boundary $\gamma(t) = 1 + u(t)$ are

$$v(t, \gamma(t)) = \dot{u}(t), \tag{16}$$

i.e. a kinematic compatibility condition. The piston force g is calculated by

$$g(t) = h(t) + \eta \frac{\partial v}{\partial x}(t, \gamma(t))$$

where h is an additional given force. For Burger's equation, an ALE-formulation is used for the moving domain and afterwards it is discretised in space using standard finite elements and in time with the implicit Crank-Nicolson scheme. For the piston equation, the implicit Newmark scheme is used. Let us denote v_{n+1} as the discrete fluid velocities and u_{n+1} as the discrete structural displacement vector at time-step $n + 1$. This leads to the discrete coupled problem

$$\begin{aligned} N(v_{n+1}, u_{n+1}) &:= M_f v_{n+1} + N_f(v_{n+1}, u_{n+1}) v_{n+1} \\ &\quad + K_f v_{n+1} - f(v_n) - C_1 u_{n+1} = 0, \\ S(v_{n+1}, u_{n+1}) &:= M_s u_{n+1} - K_s u_n - h_{n+1} \\ &\quad - C_2 v_{n+1} = 0, \end{aligned}$$

similarly as it was proposed in Eq. (4) and Eq. (5). The matrices C_1 and C_2 represent the coupling terms of this problem.

For this example, it is shown [6] that an explicit staggering algorithm is only conditionally stable whereas a fully implicit coupling algorithm is unconditionally stable.

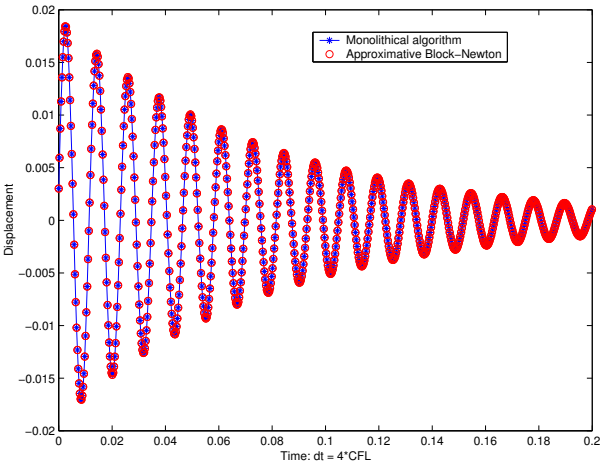


Figure 3: *Displacements of the piston (iterative method)*

In Fig. (1) we see the displacement of the piston for $\alpha = \frac{3}{2}$ and $\eta = 0.01$. If a simple staggering — half explicit time stepping in the coupling — algorithm is used, we obtain an instable response. Applying a fully implicit time stepping method, the now strongly coupled equations have to be solved jointly.

If the coupled system is solved jointly for fluid and structure variables, we see the results in Fig. (2), i.e. a stable, decaying solution.

Here a joint solution procedure (monolithical algorithm [2]) was used on the overall system of equations; in this case fairly easy because of the character of the model problem.

Now we assume that we may only use the solvers for fluid and structure separately, to simulate the situation that we have two software packages, each solving one of the two subproblems. So we can use the solution procedure for strong coupling described in the previous paragraph. The results are shown in Fig. (3), and we see the agreement with Fig. (2), i.e. the proposed algorithm computes the correct solution.

In Fig. (4), the number of solution steps (calls of the solvers for the subsystems) are shown. The approximative Block-Newton method needs almost always fewer solution steps than the Block-Gauss-Seidel method.

7 Conclusions

We have presented a new iterative method for the solution of strongly coupled fluid-structure interaction problems. The computation of these problems often requires a totally implicit formulation due to stability reasons. Hereby, large sys-

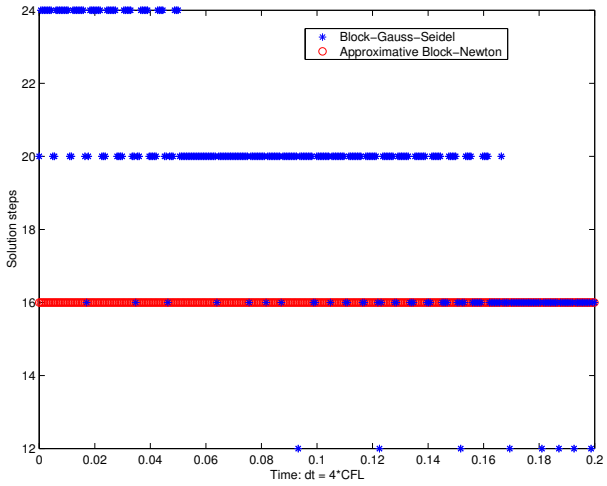


Figure 4: *Number of solution steps*

tems of nonlinear equations have to be solved. If we want to use existent solvers or software for the subsystems, often Block-Jacobi or Block-Gauss-Seidel methods are applied. As is well known from the iterative solution of linear systems, these simple methods are not always convergent. Here, we have introduced a Block-Newton method where the computation of the derivatives has been approximated by the iterative solvers for the subsystems. We have applied the approximative Block-Newton method to a one-dimensional model problem and have shown that this method is more efficient than the Block-Gauss-Seidel method.

References

[1] S. Artlich, W. Mackens: "*Newton-coupling of fixed point iterations*", in "Numerical Treatment of Coupled Systems", W. Hackbusch, G. Wittum (Eds.), Vieweg, Braunschweig, 1995

[2] F. J. Blom: "*A monolithical fluid-structure interaction algorithm applied to the piston problem*", Comp. Meth. Appl. Mech. Eng. **167**(1998)369–391

[3] R. Codina and M. Cervera, "*Block-iterative algorithms for nonlinear coupled problems*", pp. 115–134 in "Advanced Computational Methods in Structural mechanics", M. Papadrakakis, G. Bugeda (Eds.), CIMNE, Barcelona, 1996

[4] G. Dubini, R. Pietrabissi and F. M. Montevecchi: *Fluid-structure interaction in bio-fluid mechanics*, Med. Eng. Phys., **17**(8)(1995)609–617

[5] D. Errate, M. J. Esteban and Y. Maday: "*Couplage fluide-structure. Un modele simplifie en dimension 1*", C.R.A.S. Paris — Ser. 1, **318**(1994)275–281

[6] C. Grandmont, V. Guimet and Y. Maday: "*Numerical analysis of some decoupling techniques for the approximation of the unsteady fluid-structure interaction.*" Proceedings of the Second European Conference on Numerical Mathematics, Heidelberg, September 1997

[7] J. Mouro and P. Le Tallec: "*Fluid Structure Interaction with Large Structural Displacements*", Comp. Meth. Appl. Mech. Eng. **190**(2001)3039–3067

[8] S. Piperno, C. Farhat and B. Larrouturou: "*Partitioned procedures for the transient solution of coupled aeroelastic problems*", Comp. Meth. Appl. Mech. Eng. **124**(1995)79–112

[9] J. Steindorf and H. G. Matthies: "*Efficient partitioned methods for the computation of fluid-structure interaction on parallel computers*". Proceedings of the Third Euro-Conference on Parallel and Distributed Computing for Computational Mechanics. Civil-Comp Press, Edinburgh, 1999

Technische Universität Braunschweig
Informatik-Berichte ab Nr. 1998-02

1998-02	S. Petri, M. Bolz, H. Langendörfer	Transparent Migration and Rollback for Unmodified Applications in Workstation Clusters
1998-03	M. Cohrs, E. H. A. Gerbracht, W. Struckmann	DISKUS - Ein Programm zur symbolischen Diskussion reeller elementarer Funktionen
1998-04	C. Lindig	Analyse von Softwarevarianten
1998-05	Gregor Snelting, Frank Tip	Reengineering Class Hierarchies Using Concept Analysis
1998-06	Juliana Küster Filipe	On a Distributed Temporal Logic for Modular Object Systems
1998-07	J. Schönwälder, M. Bolz, S. Mertens, J. Quittek, A. Kind, J. Nicklisch	SMX - Script MIB Extensibility Protocol Version 1.0
1998-08	C. Heimann, S. Lauterbach, T. Förster	Entwurf und Implementierung eines verteilten Ansatzes zur Lösung langrechnender Optimierungsprobleme aus dem Bereich der Ingenieurwissenschaften
1999-01	A. Zeller	Yesterday, my program worked. Today, it does not. Why?
1999-02	P. Niebert	A Temporal Logic for the Specification and Verification of Distributed Behaviour
1999-03	S. Eckstein, K. Neumann	Konzeptioneller Entwurf mit der Unified Modeling Language
1999-04	T. Gehrke, A. Rensink	A Mobile Calculus with Data
2000-01	T. Kaiser, B. Fischer, W. Struckmann	The Modula-2 Proving System MOPS
2000-02	J. Saperia, J. Schönwälder	Policy-Based Enhancements to the SNMP Framework
2000-03	A. Casties	Finite-Element-Interpolation der rumlichen Dichten eines Vielteilchensystems auf ungeordneten Gittern
2000-04	J. Koslowski	A 2-dimensional view of the Chu-construction
2000-05	S. Eckstein, P. Ahlbrecht, K. Neumann	Von parametrisierten Spezifikationen zu generierten Informationssystemen: ein Anwendungsbeispiel
2000-06	F. Strauß, J. Schönwälder, M. Mertens	JAX - A Java AgentX Sub-Agent Toolkit
2000-07	F. Strauß	Advantages and Disadvantages of the Script MIB Infrastructure
2000-08	T. Gehrke, U. Goltz	High-Level Sequence Charts with Data Manipulation
2000-09	T. Firley	Regular languages as states for an abstract automaton
2001-01	K. Diethers	Tool-Based Analysis of Timed Sequence Diagrams
2002-01	R. van Glabbeek, U. Goltz	Well-behaved Flow Event Structures for Parallel Composition and Action Refinement
2002-02	J. Weimar	Translations of Cellular Automata for Efficient Simulation
2002-03	H. G. Matthies, M. Meyer	Nonlinear Galerkin Methods for the Model Reduction of Nonlinear Dynamical Systems
2002-04	H. G. Matthies, J. Steindorf	Partitioned Strong Coupling Algorithms for Fluid-Structure-Interaction
2002-05	H. G. Matthies, J. Steindorf	Partitioned but Strongly Coupled Iteration Schemes for Nonlinear Fluid-Structure Interaction
2002-06	H. G. Matthies, J. Steindorf	Strong Coupling Methods